SNMP management protocol.[103] By placing Local JiNao monitors on routers or next to routers, JiNao could respond to attacks by blocking offending packets. Furthermore, the authors planned to track the DARPA joint effort called Common Intrusion Detection Framework (CIDF) to support interoperability and module reusability with other DARPA sponsored projects.[104]

217. Thus, JiNao had the following salient features:

- Processed network packets to look for intrusive behavior and network faults.
- Ran on or next to routers and gateways.
- Designed to be independent of any particular network protocol.
- Initial implementation would analyze the OSPF routing protocol and the SNMP management protocol.
- Analyzed network traffic with both misuse and anomaly detection engines.
- Used NIDES statistical algorithms for anomaly detection.
- Correlated the results of the misuse and anomaly detectors.
- Correlated encrypted and decrypted versions of the same packet to enhance analysis.
- Responded to attacks by alerting higher-level monitors (via SNMP trap messages), modifying analysis, and blocking traffic.
- Modified analysis based on events detected by other sensors and monitors.
- Exchanged information with other monitors using a standardized network protocol (SNMP) and databases (MIB).
- Used the same misuse and anomaly detection approaches at hierarchically higher monitors.

[103] *JiNao Report* at 5-6, 14, 15.
[104] *JiNao Report* at 13-14.

- Supported peer-to-peer and hierarchical monitoring for higher-level monitors.
- Correlated and aggregated/integrated information at regional and global levels.
- Designed to integrate with additional network management or security applications by using the standards-based SNMP framework.
- Designed to support interoperability and reusability with other DARPA-sponsored projects efforts via the emerging Common Intrusion Detection Framework (CIDF).

218. Figure 8 shows a composite view of the fully scalable architecture of JiNao as described in the *JiNao Report*. It shows several "Local JiNao" systems at the bottom that monitor individual network routers and several higher-level monitors communicating in a peer-to-peer and hierarchical architecture. The MCNC/NCSU contract with DARPA for which this report was written only required them to implement the local JiNao components.[105]

219. In the lower left portion of Figure 8 is a detailed description of a local JiNao system for a single router. At the bottom of this local JiNao system is the Intercept/Redirect module (1) which intercepted a network packet before the router's software could process it, forwarded the packet to the analysis modules (2) (3) (4), and if (2) did not complain, would forward the packet to the router's software. If JiNao determined that a packet was part of an attack, this module could respond by preventing the malicious packets from being sent to the router's software.

220. The Intercept/Redirect module forwarded the packet to the next module in the JiNao system, the Simple Misuse Detection module (2). The *JiNao Report* actually calls this the rule-based "prevention module," and it applied simple rules to a single packet in order to quickly detect clear security violations.

[105] *JiNao Report* at 3.

221. The Simple Misuse Detection module (2) forwarded the packets to the local detection module that consisted of an anomaly detector (3) and a misuse detector (4) that detected attacks that spanned multiple packets. The anomaly and misuse detectors forwarded their results to the Local Decision Module (5), which correlated information from the other sensors and determined whether a response should be taken. Potential responses included directing the Simple Misuse Detector (5) to start blocking certain packets, adjusting the analysis of the misuse and anomaly detectors, making the analysis available to other monitors through a MIB (6), and alerting high-level monitors through an SNMP trap message.

222. A local JiNao system communicated with a hierarchically higher monitor through the network agent (7) using a standardized SNMP protocol (8). JiNao's local results and configuration information were available to higher-level monitors through a Management Information Base (MIB) (6), and alerts were automatically sent to higher-level monitors though SNMP trap messages, part of the standardized SNMP network management architecture.

223. The JiNao architecture supported hierarchically higher monitors called Remote Management Applications (9) that used the same anomaly and misuse algorithms (10) as the local JiNao monitors. Remote Management Application monitors could communicate peer-to-peer with other Remote Management Application monitors (11) via the standard ManagerToManager portion of the SNMP protocol to aggregate results to support regional detection. Furthermore, Remote Management Application monitors could communicate hierarchically (12) to develop a more global view of potential misuse in the network. Although the *JiNao Report* states that a hierarchical system had not yet been created, the report is clear that the architecture described was suitable for implementing such a hierarchy. Furthermore, because the *JiNao Report* explicitly directed one of skill to use the known hierarchical architecture of the SNMP framework,

one of skill in the art would have understood this reference to disclose and enable hierarchical JiNao monitors.

224. JiNao processed network packets. The messaging infrastructure for internal JiNao modules used a generic Protocol Data Unit (PDU), so they could be tailored for any protocols. MCNC/NCSU proposed to deliver solutions for the OSPF and SNMP protocols, and the *JiNao Report* mentions the possibility of using JiNao to process HTTP packets to protect web servers. In addition, the *JiNao Report* explicitly states that JiNao monitored data volume.[106]

225. JiNao analyzed network traffic through a combination of anomaly detection based on the NIDES statistical algorithms and misuse detection. The Simple Misuse Detection module (labeled (2) in Figure 8 and called "Prevention Module" in the report) used simple rules that could quickly evaluate individual packets for clear security violations. The Stateful Misuse Detection Module (labeled (4) in Figure 8 and called "Protocol Analysis" in the report) used a collection of Finite State Machines (FSMs) to detect more complex attacks that can span multiple packets. The anomaly detector (3) was based on the NIDES statistical algorithms.

226. The Protocol Analysis module was a collection of finite state machines (FSM). Each FSM was a signature designed to detect a specific attack. UC Davis had also done research on methods to detect attacks through protocol analysis, and thus I am familiar with this type of analysis. Typically, such analysis sets forth a set of rules describing the "correct" behavior of a particular protocol, and then looks for deviations from this known proper behavior. Such analysis may also look for known exploitations of known vulnerabilities of a particular protocol.

[106] *JiNao Report* at 19.

227. The Local Decision Module (labeled (5) in Figure 8) correlated the results from the anomaly and misuse detectors. Hierarchically higher monitors would use similar detection/analysis functions to "correlate intrusion events among several routers."[107] For example, as described in the *JiNao Report*, JiNao could correlate conflicting information from multiple JiNao systems to determine that a router between those two systems (Bob) was bad.[108]

228. JiNao could respond to suspicious events in several ways. First, the Local Decision Module (5) could update the MIB Database (6), thus making the information available to hierarchically higher monitors. Second, the Local Decision Module could immediately automatically alert hierarchically higher monitors with an SNMP trap message. Third, the Local Decision Module could direct the local analysis modules to modify their analyses. Fourth, the Local Decision Module could raise an alert to the security officer via an email or a graphical user interface. Fifth, the Local Decision Module could interact directly with the router to modify its settings. Sixth, the Simple Misuse Detection module (labeled (4) in Figure 8 and called "Prevention Module" in the document) could direct the Interception/Reception module (1) not to forward the packet to the router's software.

229. The JiNao modules would satisfy the "monitor" limitation under any of the parties' definitions of that term.

230. In my opinion, as more fully reflected in Exhibit G to my report, the JiNao system disclosed in the *JiNao Report* and related slides satisfied every limitation of the asserted claims of the patents-in-suit (except for '338 claims 2-3, 5-8 and 10) and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged

[107] *JiNao Report* at 13.
[108] *JiNao Report* at 35.

inventions claimed in the patents-in-suit. Consequently, the noted claims of the patents were not novel when filed.

231. In addition, as reflect in Exhibit G, all of the remaining asserted claims of the patents-in-suit are invalid due to obviousness.

232. It would have been obvious to one of ordinary skill in the art to use JiNao's statistical profile-based anomaly detection of network packets to analyze the additional network traffic data categories analyzed and described by: "A Network Security Monitor—Final Report," NetRanger, ISS RealSecure, Network Flight Recorder, Network Security Probe, Gabriel, synkill, SNMP/RMON IETF standards, and the SunScreen Firewall.[109] This is particularly true since the JiNao Report explicitly mentions monitoring SNMP.[110] As shown in these references, all of these network traffic data categories were well-known, and it was also well-known that these categories could be analyzed using either signature detection, statistical profile-based anomaly detection, or both. One of skill would have been motivated to combine JiNao with these additional references in order to monitor additional types of network traffic for improved detection capabilities.

233. In addition, to the extent there were any differences between the configuration of JiNao and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of JiNao based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

[109] *See* the NSM chart at Ex. C for the full titles and dates of all these references.

[110] *JiNao Report* at 5-6, 15, 17, 18

6. EMERALD 1997

234. This section covers the EMERALD design as described in the paper titled "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," published in the Proceedings of the 20th National Information Systems Security Conference in October of 1997 ("*Emerald 1997*"). Phil Porras, one of the named inventors of the patents-in-suit, and Peter Neumann are the authors of this paper. The primary sponsor for this work was the Department of Defense via the Defense Advanced Research Projects Agency (DARPA).

235. *Emerald 1997* describes an architecture for monitoring an enterprise network collectively called EMERALD. EMERALD analyzed an input stream such as network packets, and applied anomaly detection, misuse detection, correlation, and aggregation to detect possible misuse. EMERALD's internal modules used a well-defined API to pass messages between modules; EMERALD also used a well-defined protocol to support hierarchical and peer-to-peer communications to provide scalability and detect larger scale attacks.

236. EMERALD as described in this publication had the following salient features:

- Monitored network services and network components such as routers and gateways.
- Analyzed network packets, audit logs, application logs, and results of other intrusion detection sensors; these data sources are referred to as an event stream.
- Used anomaly detection based on the NIDES algorithms to analyze the event stream.
- Used signature detection to analyze the event stream for known examples of misuse.
- Provided an Application Programming Interface (API) to allow third-party modules and sensors to participate in an EMERALD system.

- Correlated analysis from the anomaly detection engine, the signature detection engine, and potentially other analysis engines.
- Supported hierarchical monitoring beginning at the lowest level (service monitor), with the next higher level being a domain monitor, and the highest level being an enterprise monitor.

237. The system described in *Emerald 1997* shared the basic elements of SRI's previous work in IDES and NIDES, but it was designed to be distributed and independent of any particular data source. The paper states that EMERALD's distributed design addresses scalability. Designing the system to be independent of the underlying data source makes it easier to incorporate other data sources, including results from other intrusion detection sensors.

238. EMERALD took the basic NIDES architecture – an anomaly engine, a signature/misuse engine, and a resolver – and generalized it for any input stream. The earlier SRI intrusion detection systems (IDES, NIDES and Safeguard) shared a similar architecture, but they were largely focused on an input stream of host-generated data (e.g., audit trails) to analyze host-related subjects (e.g., user accounts and processes). EMERALD generalized the approach so that the system was not tied to any specific input stream or subject matter.[111]

239. Furthermore, because the input stream was generalized, the output event stream for one monitor could be used as the input stream for another monitor. Because EMERALD monitors could be composed in this manner, a hierarchy of monitors could be created, and this hierarchical approach in turn created the ability to scale the analysis to larger environment.

240. Figure 9 shows a composite view of the fully scalable architecture of the EMERALD system as described in *Emerald 1997*.

[111] *Emerald 1997* at Fig. 1 shows the "generic EMERALD monitor architecture."

241. In the lower left portion of Figure 9 is a description of an EMERALD "Service Monitor" (1). It read input streams such as network packets (2), and then analyzed that input stream with an anomaly detection engine (3) and a signature detection engine (4). The results of these analyses engines were passed to the Resolver (5), which could correlate the results, take some type of response (6), or pass its analysis to hierarchically higher monitors (7). The boxes to the right of the annotated Service Monitor (1) are simply other Service Monitors.

242. The next level of monitor in the hierarchy was called the "Domain Monitor" (8), and it analyzed and correlated the results (7) from the lower level Service Monitors (1). The Domain Monitor shared the same architecture as the Service Monitor, including an anomaly detection engine, a signature/misuse detection engine, and a Resolver. The Domain Monitor could also pass the results of its analysis to other Domain Monitors (9) via peer-to-peer sharing, or it could pass the results to a hierarchically higher Enterprise Monitor (10).

243. The top level of monitor in the hierarchy was called the "Enterprise Monitor" (11). The Enterprise Monitor analyzed and correlated the Domain Monitors' results (10). The Enterprise Monitor shared the same architecture as the Domain and Service Monitors, namely, an anomaly detection engine, a signature/misuse detection engine, and a resolver.

244. EMERALD processed any type of input stream. Examples in the *Emerald 1997* paper included audit data, network datagrams (i.e., packets), SNMP traffic, application logs, and results from other intrusion detection sensors.[112]

245. EMERALD used a combination of anomaly detection, signature detection, and a resolver to analyze the input streams. The signature/misuse engine detected known

[112] *Emerald 1997* at 356.

intrusive behavior. The anomaly engine detected unusual activity that might indicate previously unknown intrusive behavior. The resolver integrated and correlated information from the anomaly and signature engines and potentially from other monitors as well.

246. The signature analysis engine is described as mapping an incoming event stream against "abstract representations of event sequences that are known to indicate undesirable activity."[113] The objectives of the signature analysis depend on which level in the hierarchy the analysis is operating at: service monitors target known attacks on network services and infrastructure, whereas above the service layer, the signature engines scan aggregated reports for more global coordinated attack scenarios.

247. The anomaly detection engines are described as using the statistical profiling type of anomaly detection developed in NIDES. The paper notes that the "underlying mechanisms" of NIDES are "well suited to the problem of network anomaly detection, with some adaptation."[114] *Emerald 1997* then described the modifications to the statistical profiling mechanism needed in order to achieve the generality of input desired in EMERALD.

248. EMERALD's resolver was responsible for responding to suspicious behavior. The paper describes several different actions the resolver could take in response to detected suspicious activity including: (1) closing connections, (2) terminating processes, (3) calling a host's integrity checker to verify operating state, (4) propagating information to other monitors, (5) modifying the analysis of its detection engines, and (6) sending information to the user interface.[115]

[113] *Emerald 1997* at 359.
[114] *Emerald 1997* at 359.
[115] *Emerald 1997* at 361.

249. The common specification of the patents-in-suit is remarkably similar to the text of the *Emerald 1997* paper. In order to assist my comparison of these two documents, I have reviewed a highlighted comparison of the two documents (see Ex. GG).[116] My review of this comparison indicates that a great deal of the text of *Emerald 1997* was reproduced verbatim or nearly verbatim in the specification of the patents-in-suit. In addition, Figures 1 and 2 of *Emerald 1997* are virtually identical to Figures 2 and 3 of the patents. I understand that one of the inventors, Mr. Porras, described *Emerald 1997* as merely an "early conceptual design."[117] This position is not plausible in light of the substantial overlap between the patents and this paper. I have examined the portions of the specification not present in the *Emerald 1997* paper and I do not believe that they provide any important disclosure required to implement the system described beyond what *Emerald 1997* disclosed in light of the knowledge of one of ordinary skill in the art.

250. Furthermore, given the similarity in language between *Emerald 1997* and the patents, my opinions on invalidity and obvious with regard to *Emerald 1997* and all other NIDEs and Emerald –related references do not change regardless of which claim construction position is ultimately adopted by the Court.

251. SRI has claimed that *Emerald 1997* does not teach the specific categories of network traffic data called out in, for example, '338 claim 1, '203 claim 1, and '615 claim 1.[118] First, it is important to point out that monitoring these particular types of "measures" of network packets or "network traffic data" was not new or novel – all of

[116] This comparison chart was prepared for demonstrative purposes only, and is not intended to be a strict word-for-word comparison. Minor differences in highlighted language may exist.
[117] Porras Tr. 433.
[118] *See* SRI International, Inc.'s "Amended" Response to Symantec's Invalidity and Inequitable Conduct Contentions.

them had been monitored by other systems before. The inventors have acknowledged this fact for many of the claimed categories.[119]

252. Second, as explained in the Expert Report of Frederick Avolio, *Emerald 1997* explicitly stated that the disclosed EMERALD system could monitor and analyze "network infrastructure" including firewalls. *Emerald 1997* also disclosed monitoring an event stream from an application log, which would include a firewall log. Firewalls in 1997 provided a common set of monitoring and logging features which were well-known in the art at the time. Thus, one of skill would have understood from the *Emerald 1997* disclosure that one should monitor network connections, including network connection requests and denials, and data transfers, including network packet data volume/network packet data transfer volume. I have reviewed Mr. Avolio's report and spoken with him in detail about it. I agree with and adopt in my own report his report, including his analysis and conclusions. I also adopt the references he relied upon in reaching his conclusions.

253. In addition, *Emerald 1997* disclosed monitoring an event stream of SNMP traffic[120] and monitoring network infrastructure.[121] As explained more fully in my discussion of HP Openview and the Internet standards (RFCs), SNMP traffic and related network infrastructure management data provides monitoring for a variety of different categories of network traffic data. The claimed categories of network traffic data would have been understood by one of ordinary skill in the art to be disclosed by this explicit reference to monitoring SNMP traffic and network infrastructure. See also my table at Exhibit X.

254. Furthermore, the types of network traffic one should monitor to detect suspicious activity or network intrusions flow naturally from the types of attacks that one

[119] Porras Tr. 289-295, 444-454; Valdes Tr. 283-287.
[120] *Emerald 1997* at 356.
[121] *Emerald 1997* at 355.

is looking for. As the intrusion detection and computer security fields developed, practitioners in the field began cataloging and tracking security-related intrusions seen by different computer networks. For example, after the Morris worm incident in November 1988, which caused extensive damage to different internet systems, DARPA called on the Software Engineering Institute at CMU to set up a center to coordinate communication among experts during security emergencies.[122] Known as the Computer Emergency Readiness Team ("CERT"), CERT issued public advisories to warn of new attacks or vulnerabilities.

255. For example, in September 1996, CERT first issued a warning about TCP SYN flooding and IP Spoofing attacks.[123] This advisory explained the SYN flooding attack, in which numerous requests to open a TCP connection are sent that cannot be responded to, flooding the system with half-open connections and making it difficult for the system to continue to communicate. One of skill in the art at the time would have known about SYN flooding attacks and understood that one should monitor network connection requests and denials to detect such an attack. A variety of other types of network attacks were also well-known at the time, which would have similarly indicated different categories of network traffic to monitor. [124]

256. To the extent that the various categories of network traffic data or measures of network packets are not disclosed in *Emerald 1997*, it would have been obvious to one of skill in the art to combine *Emerald 1997* with any of the many well-known firewalls, intrusion detection research systems, commercial intrusion detection

[122] http://www.cert.org/meet_cert/meetcertcc.html

[123] CERT Advisory CA-1996-21 [SYM_P_0548726-734]. *See also* Schuba et al., "Analysis of a Denial of Service Attack on TCP," Proc. of the 1997 IEEE Symposium on Security and Privacy, Oakland, CA, 208-23 (May 4-7 1997) [SYM_P_0535408-28].

[124] Numerous other attacks were well-known at the time. *See, e.g.*, CERT Advisory CA-1996-26 "Denial-of-Service Attack via ping, Dec. 18, 1996 [SYM_P_0548718-725] (discussing ICMP error packets and oversized ICMP datagrams).

systems, and IETF defined standards on what to monitor for network infrastructure, all of which were already monitoring these network traffic data categories. In fact, *Emerald 1997* explicitly pointed the reader to other systems, including NSM.[125] For example, both ISS RealSecure and NetRanger were well-known network intrusion detection systems, and one of skill upon reading *Emerald 1997* would have been motivated to look at the traffic data being monitored by such systems in order to select the best categories of traffic for use with EMERALD. Indeed, I understand that the inventors in 1997 were actually working with and looking at Sun Microsystem's SunScreen firewall, further supporting my opinion that firewalls and other IDS systems would have been obvious sources of information about useful network traffic data categories.[126] Since the SunScreen firewall and NetRanger also included the use of VPNs, it similarly would have been inherent or obvious to combine the system of *Emerald 1997* with the VPNs in use by these products.

257. I also understand that SRI claims that the *Emerald 1997* paper does not provide an "enabling disclosure" of a statistical detection method.[127] I disagree. *Emerald 1997* included an entire section entitled "Scalable Profile-Based Anomaly Detection" which describes how EMERALD will incorporate and modify the well-known NIDES algorithms for statistical profiling. *Emerald 1997* also pointed the reader to additional references on the NIDES algorithms. Furthermore, other additional SRI publications had also already disclosed the NIDES algorithms in detail.[128] To the extent that the specific

[125] *Emerald 1997* at 364.

[126] Valdes Tr. 68-69; 12-123. *See also Emerald – Conceptual Design 1997* at p. 88. [SRI 012400].

[127] *See* SRI International, Inc.'s "Amended" Response to Symantec's Invalidity and Inequitable Conduct Contentions.

[128] *See, e.g.*, A. Valdes and D. Anderson, *Statistical Methods for Computer Usage Anomaly Detection Using NIDES,* Proc. of the Third International Workshop on Rough Sets and Soft Computing, January 1995 ("*Statistical Methods*") [SYM_P_0068937-942]. Mr. Valdes testified that the equations disclosed in this paper were suitable for network monitoring. Valdes Tr. 344-346. Furthermore, the patents' specification explicitly states

algorithms for statistical profiling are not incorporated-by-reference into *Emerald 1997*, it would have been obvious to one of skill in the art to combine *Emerald 1997* with prior NIDES-related publications, since *Emerald 1997* states that the NIDES algorithms form the basis for the statistical profiling disclosed. One of skill would have been motivated to further investigate these specific algorithms in order to improve the performance of the system.

258. In my opinion, as more fully reflected in Exhibit H to my report, the system disclosed in *Emerald 1997* satisfied every limitation of each of the asserted claims of the patents-in-suit (except '338 claim 7) and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the inventions claimed in the patents-in-suit. Consequently, the noted claims of the patents-in-suit were not novel when filed.

259. In addition, as reflected in Exhibit H, the remaining indicated claim is invalid as obvious, for the reasons detailed above.

a. CMAD and Conceptual Overview

260. In addition to the *Emerald 1997* paper, SRI also published two shorter earlier descriptions of the EMERALD system. *See* P. Neumann, P. Porras and A. Valdes, "Analysis and Response for Intrusion Detection in Large Networks," Summary for CMAD Workshop, Monterey, 12-14 November 1996 ("Emerald CMAD") and P. Porras and P. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances Conceptual Overview," December 18, 1996 ("Emerald – Conceptual Overview"). While neither of these references is as detailed as *Emerald 1997*, both also anticipate or render obvious many of the asserted claims of the patents-in-suit for the reasons stated above.

that the techniques described in this paper may be used for the "profile engine" which "can profile network activity via one or more variables called measures." '338 col. 5:43-50.

b. Live Traffic Analysis and Conceptual Design 1997

261. SRI also published two additional papers regarding the EMERALD system: P. Porras and A. Valdes, "Live Traffic Analysis of TCP/IP Gateways," http://www.sdl.sri.com/projects/emerald/live-traffic.html, Internet Society's Networks and Distributed Systems Security Symposium, Nov. 10, 1997 ("*Live Traffic Analysis*") and P. Porras and P. Neumann, "Conceptual Design and Planning for EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," Version 1.2 20 May 1997 ("Emerald - Conceptual Design 1997"). I understand that there is a debate over whether or not these documents were publicly available prior to November 9, 1997. If public, *Live Traffic Analysis* anticipates all of the asserted claims of all of the patents-in-suit, except for '338 claim 25, which is rendered obvious, as indicated in Exhibit L. If public, Emerald – Conceptual Design 1997 anticipates or renders obvious all of the asserted claims of the patents-in-suit, for the reasons discussed above and those noted on Exhibit K.

c. Emerald 1997, Intrusive Activity 1991, NIDES 1994

262. *Emerald 1997* also references two additional publications in both the text of the paper itself, and in the list of references. *Emerald 1997* explains that the statistical algorithms in H. Javitz and A. Valdes, "The NIDES statistical component description and justification," Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA March 1994 ("*NIDES 1994") provides the foundation for the profile-based anomaly detection in *Emerald 1997*:

> "Requirements for an anomaly-detection system that became IDES were documented in [6]. This research led to the development of the NIDES statistical profile-based anomaly-detection subsystem (NIDES/Stats), which employed a

> wide range of multivariate statistical measures to profile the behavior of individual users [9]."[129]

263. In addition, *Emerald 1997* also directs one to the Network Security Monitor (NSM) system for analysis of network traffic, and identifies L.T. Heberlein, B. Mukherjee, K.N. Levitt., "A Method to Detect Intrusive Activity in a Networked Environment," Proc. 14th National Computer Security Conference, pp. 362-371, Oct. 1991 ("*Intrusive Activity 1991*") as describing NSM:

> "[T]he Network Security Monitor [7] seeks to analyze packet data rather than conventional audit trails..."[130]

264. Given these explicit references in *Emerald 1997* to both *NIDES 1994* and *Intrusive Activity 1991*, these three references should be considered to be a single disclosure for purposes of anticipation. In the alternative, the citations above provide a motivation to combine these three references in order to make and improve the statistical profiles and network traffic monitoring claimed in the patents-in-suit. As shown in Exhibit U, *Emerald 1997*, *NIDES 1994* and *Intrusive Activity 1991* together satisfied every limitation of the asserted claims of the patents-in-suit (except claim 7, which is invalid as obvious) and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in the patents. Consequently, the noted claims of the patents were not novel when filed.

7. Network NIDES

265. This section covers NIDES and future design work (collectively called "Network NIDES" in this section) as described in the report titled "Next-generation Intrusion Detection Expert System (NIDES) A Summary," dated May of 1995 ("*Network NIDES*"). SRI International performed this work, with Debra Anderson, Thane Frivold, and Alfonso Valdes (one of the named inventors) listed as the authors of this report. The

[129] *See Emerald 1997* at 359.
[130] *See Emerald 1997* at 364.

primary sponsor for this work was the Department of the Navy, Space and Naval Warfare Systems Command (SPAWAR).

266. NIDES was under development from about 1992 through late 1994, with this report delivered in 1995, although NIDES itself was essentially an extension to SRI's Intrusion Detection Expert System (IDES), which dates back to the mid-late 1980s. NIDES was designed to detect intrusive behavior using a combination of anomaly and misuse detection. NIDES was essentially a redesign of the SRI's earlier IDES, with a focus on a modular and more flexible approach. This modularization and flexibility also helped retarget NIDES to other platforms than its original Solaris audit-based analysis. For example, NIDES was modified to support (1) monitoring of Trusted Xenix systems (Advisor Project), (2) database transaction logs (FBI FOIMS-IDES Project), (3) and analyzing applications (Safeguard Project).[131]

267. The basic NIDES architecture included an anomaly engine performing statistical profiling, a signature/misuse engine, and a resolver. The anomaly detection engine detected previously unknown intrusive activity based on the assumption that this behavior would appear different than normal behavior. The signature/misuse detection engine detected known intrusive activity. The resolver received the analysis from the anomaly and misuse detection engines, integrated multiple reports associated with the same underlying event, and responded by sending a message to the user interface or sending out email.[132]

268. Network NIDES had the following salient features:

- Monitored network components such as routers and gateways.

[131] *Network NIDES* at 3.
[132] *Network NIDES* at 22-23.

- Analyzed network packets, audit logs, application logs, and results of other intrusion detection sensors; these data sources were referred to as an event stream.

- Used anomaly detection based on the previously published NIDES algorithms to analyze the event stream.

- Used signature detection to analyze the event stream for known examples of misuse.

- Provided an Application Programming Interface (API) to allow third-party modules and sensors to participate in the system.

- Integrated analysis from an anomaly detection engine, a signature detection engine.

- Supported hierarchical monitoring with domain NIDES monitors reporting to higher-level NIDES monitors.

269. The input stream processed by NIDES was somewhat generalized with 10 general audit record formats supported, another 90 potential types reserved, and 50 allowable user-defined audit types. The primary audit types that SRI worked with were SunOS BSM version 1, SunOS C2, and UNIX accounting records. SRI provided a small application (agen) to allow users to convert their own data sources into a format for NIDES.

270. Figure 10 shows a composite view of the NIDES system as described in *Network NIDES*. This architecture is based on what was described as implemented as of the time of this report as well as what was planned as described in Section 3 "Future Directions."

271. In the lower left portion of Figure 10 is a description of the basic NIDES monitor responsible for a particular domain (1). It would read an input stream such as network packets (2), and then analyzed that input stream with an anomaly detection engine (3) and a signature detection engine (4). The results of these analyses engines were passed to the Resolver (5), which could integrate the results, take some type of response such as alerting a user (6), or pass its analysis to hierarchically higher monitors

(7). The boxes to the right of the Domain NIDES monitor (1) are simply other Domain NIDES monitors.

272. Above the local domain NIDES monitors is a "higher-level NIDES" monitor (8) that would process the results from the lower level Domain NIDES monitors (1). This higher-level NIDES monitor would be "responsible for the network that supports all the local domains."[133]

273. New rules would be developed specifically to detect "suspicious network activity."[134] Also, the statistical component would be "easily adaptable to network intrusion detection"[135] by adapting the subjects (e.g., hosts instead of users) and measures for those subjects (e.g., "the amount of traffic originating or being received" by the subjects).[136]

274. NIDES processed any type of input stream as long as it is mapped to the NIDES audit format via an application called "agen." In addition to those data sources that were implemented for NIDES, the report also suggested input streams could consist of network packets, reports from other NIDES monitors, "a series of connection attempts," "the amount of traffic," reports from the Network Intruder Detector (NID), reports from TCP wrappers about connection attempts, reports from firewalls, SNMP, and reports from application gateways.[137]

275. Furthermore, *Network NIDES* monitors would be considered monitors under any of the parties' claim constructions.

[133] *Network NIDES* at 31.
[134] *Network NIDES* at 31.
[135] *Network NIDES* at 32.
[136] *Network NIDES* at 31.
[137] *Network NIDES* at 17, 32, 33.

276. In my opinion, as more fully reflected in Exhibit K to my report, the system disclosed in *Network NIDES* satisfied every limitation of the indicated claims of the patents-in-suit and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in the patents. Consequently, the noted claims of the patents were not novel when filed.

277. In addition, as reflect in Exhibit K, the remaining asserted claims of the patents-in-suit are invalid due to obviousness.

278. In particular, it would have been obvious to combine the *Network NIDES* system with other hierarchical intrusion detection systems such as DIDS and GrIDS, and to use their correlation capabilities in *Network NIDES.* It also would have been obvious to combine *Network NIDES* with other systems analyzing network packets and different categories of network traffic data, and to perform NIDES statistical profiling on these categories.

279. In addition, to the extent there were any differences between the configuration of *Network NIDES* and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of *Network NIDES* based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

8. Frank Feather Thesis

280. This section covers Frank Feather's dissertation as described in his thesis "Fault Detection in an Ethernet Network via Anomaly Detectors," dated May of 1992 ("*Feather Thesis*"). This dissertation was based on Frank Feather's research at Carnegie Mellon University and extended the Performance and Anomaly Monitoring System (PAMS) developed primarily by Pamela Hayes. Hayes' work on PAMS was published

as a Senior Thesis at Chatham College, Pittsburgh, PA in May of 1987 ("Characterizing Numeric Time Series Data: Applications to Ethernet LAN Data"). The *Feather Thesis* covered work up through May of 1992. The original PAMS work was published in 1987.

281. The *Feather Thesis* describes a network monitoring system that used an algorithm that was very similar to the Haystack statistical intrusion detection algorithm. The work's primary focus was to quickly detect a wide range of network faults, including those generated by a number of malicious or intrusive activities. The system performed fault detection by applying anomaly detection to a number of measured features of the network traffic. An anomaly was determined by comparing current measurements of network activity to historical measurements for the same measurements; the historical measurements were regularly updated using an exponential aging factor like that used in NIDES. Once anomalous activity was detected, the system further refined the analysis by applying a fault feature vector to the anomaly vector in order to determine the likely underlying cause of the detected anomalous activity.

282. Feather's dissertation had the following salient features:

- Monitored network packets.
- Designed to detect faults in the network ranging from a wide range of causes, including network worms, inappropriate use of the network, mischievous users, or malicious network misuse (e.g., connecting to the network without permission).
- Extended an earlier anomaly detection system called "Performance and Anomaly Monitoring Systems" (PAMS).
- Added the Address Anomaly Detection System (AADS) to extend PAMS' network-wide analysis to host and host-to-host analysis.
- Applied profiles of past activity to determine if currently observed network activity was anomalous.
- Updated profiles using the same exponential decay approach used by SRI's NIDES algorithm.

- Applied a fault feature vector to help determine the cause of the detected anomalous activity.
- Discussed a wide range of network features to measure.
- Deployed and evaluated on an Ethernet network analyzing packets including TCP/IP packets.
- Closely resembled the Haystack intrusion detection system algorithm.
- Suggested also applying the analysis to routers, bridges, file servers, and other network components.

283. The *Feather Thesis* addressed the problem of quick detection and diagnosis of soft failures in a network before they became a noticeable problem to the users. "Soft failures" are generally characterized by degraded performance of the network. In computer security lingo, this is generally referred to as denial of service. "Hard failures" are a complete failure of a component of the network so that no packets can be delivered. Unaddressed soft failures may eventually lead to hard failures. For example, a failing component in the network may lead to intermediate packet loss (soft failure), but that failing component may eventually completely fail leading to hard failure.

284. There are many potential reasons for soft failures in the network. Some of the examples of causes for soft failure listed in the dissertation include inappropriate use of the network, temporary congestion causing delayed transmission, failed host hardware, failure of higher level protocols, mischievous users, Internet worms, or a user connecting a host to a network without permission.

285. This system described in the *Feather Thesis* measured a wide range features about the network as a whole, individual hosts, and pair-wise host communications. Examples of such measurements include packets per minute, load, broadcast packets, packet size, packet type distribution, unexpected hosts on the network, unexpected communication between pairs of hosts, and a wide range of error events.

Such error events include collisions, checksum errors, undersized packets, oversized packets, and dropped packets.

286. Once measurements for the subjects were identified, the system developed expectations of the values for these numbers; that is, it determined the normal values for each of the measurements. The normal values were determined by analyzing past behaviors for the measurements, and the expected normal values were updated regularly using an exponential smoothing of past activities such that recently measured activity were given more weight than past activities – exponential aging.

287. Once profiles of normal values were established, the system compared current measurements to historically normal values for each measurement and identified anomalous measurements.

288. To go beyond simply identifying anomalous activity, the system applied fault feature vectors to the set of anomalous features to automatically identify the likely underlying cause of the observed anomalous behavior. For example, a bridge going down would exhibit one set of anomalous features, while a host using too much memory and paging out memory to a remote disk would exhibit a different set of anomalous features. Fault feature vectors tuned to different fault types could help disambiguate the cause of the detected anomalies.

289. Figure 11 shows the basic algorithm described in this dissertation. The box on the left represents the analysis performed as part of Frank Feather's dissertation (1). At the bottom is the data source, packets, which are fed into the system (2). From the packets a description vector was computed to represent the currently observed network activity. This description vector was composed of two parts: a series of features representing system-wide performance (X_1, X_2, ..., X_m) (3), and a set of features for individually observed hosts or communicating pairs of hosts (H_1...H_k) (4). Together,

these measurements represented the description vector, which is a short-term profile of recent activity.

290. Next, code compared (5) the current description of the network traffic with the profiles (6) of past network traffic. The result of this comparison was an anomaly vector (A_1, A_2, ..., A_n) (7) describing which of the measured features were anomalous. Once anomalous activity was detected, the system tried to determine the underlying cause of the anomalous activity by applying (8) a weighted fault vector (9) for different fault types to the anomaly vector. The result of this analysis is the "matching score" (10).

291. Interestingly, the approach described in Frank Feather's dissertation is strikingly similar to Haystack Labs' Haystack Intrusion Detection System (11).[138] In Haystack, the primary data source was a system's audit records (12), and a feature vector for each user was computed (13). An anomaly detection function (14) compared the current feature vector to a profile of historical behavior (15) to determine if there were any anomalous features. The result was an anomaly vector (16) that Haystack called the bernoulli vector. To further determine what type of malicious activity might be causing the detected anomalous activity, Haystack applied (17) a weighted intrusion vector (18) for different classes of intrusive activity, and the result was a weighted intrusion score (19) for each class of intrusion. Haystack took one additional step (not shown) to determine the probability of a given session having a given weighted intrusion score or higher.

292. Under Symantec's alternative claim construction, the *Feather Thesis* would satisfy the requirement that the long-term statistical profile was an exponentially

[138] I am familiar with the Haystack statistical algorithm because I analyzed Haystack, including the source code, in order to perform the analysis in my 1994 paper: B. Mukherjee, L.T. Heberlein, K.N. Levitt., "Network Intrusion Detection," IEEE Network, Vol. 8 No. 3, pp. 26-41, May/June 1994.

aged probability distribution of historical values. However, Feather's short-term statistical description was not aged.

293. The *Feather Thesis* used network packets as its primary data source. The Address Anomaly Detection System (AADS) element of the dissertation, which analyzed hosts and communicating pairs of hosts, processed data coming from the commercial Ethernet-Adapter-Karte (EAK) Monitoring System (EMS) – a network monitoring card that captured the packets and collected the primary statistics.

294. Most individual measurements were analyzed using the Performance and Anomaly Monitoring System (PAMS), which was focused on analyzing univariate time series data. For any given measurement at any given time during the day, a mean and standard deviation were computed. Newly observed data measurements that exceeded the mean by some amount (e.g., more than three standard deviations above the mean) were flagged as anomalous. Furthermore, the AADS component also reported the sudden appearance of a new network address or the sudden drop out of a usually talkative network address.

295. Once anomalous activity was detected, a fault feature vector was applied to the anomaly vector in an attempt to disambiguate the underlying cause of the detected anomalous activity. The dissertation also mentioned the need to automatically correct detected faults, but it did not go into detail as to how that might be achieved.

296. In my opinion, as more fully reflected in Exhibit L to my report, the system disclosed in *Feather Thesis* satisfied every limitation of the indicated claims of the '338 patent, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in the '338 patent. Consequently, the noted claims of the '338 patent were not novel when filed.

297. In addition, as reflect in Exhibit L, the remaining asserted claims of the '338 patent are invalid due to obviousness.

298. The *Feather Thesis* related to the field of network monitoring, which was undergoing significant expansion in the 1992 timeframe in which the thesis was written. By spring of 1993, a standard had been developed for doing hierarchical monitoring. Furthermore, the *Feather Thesis* explicitly directed the reader to use SNMP: "[w]ith SNMP now available on many platforms, it would be very easy to do anomaly detection on data from other network components."[139] It would have been obvious to combine the work of the *Feather Thesis* with hierarchical monitoring systems, because the entire network monitoring field was already moving in this direction. In addition, the network monitoring SNMP standards for the purpose of network management documented a wide range of network traffic data categories to monitor. It would have been obvious to combine the *Feather Thesis* with these additional known categories of network traffic in order to improve and expand upon its network monitoring capabilities.

299. In addition, to the extent there were any differences between the configuration of the system described in the *Feather Thesis* and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of the *Feather Thesis* based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

9. HP Open View and RFCs

300. This section covers the field of network management and in particular the HP OpenView network management platform and the Internet Engineering Task Force (IETF) standards such as SNMP, MIB, and the Remote Network Monitoring (RMON)

[139] *Feather Thesis* at 172.

specifications (RMON I and II) (collectively the "RFCs")[140]. The RFCs define an infrastructure and monitoring capability that can monitor a wide range of measurements for many different network elements, determine if those measurements are unusual, and if so, send alerts to higher-level monitors. The SNMP architecture supported a hierarchical design. Furthermore, network management platforms such as HP OpenView provided a wide range of responses for unexpected behavior, including alerting a user through graphical means, audible means, and sending messages to their pagers.

301. Network management applications such as HP OpenView and the RFCs were not classic intrusion detection systems, but they certainly performed "network monitoring" as claimed in the patents-in-suit. Furthermore, the computer security field is simply a part of the overall network management field.[141] Many network faults and problems create suspicious behavior patterns that are important for intrusion detection systems to be aware of. Many different intrusion detection systems explicitly relied upon network management systems for part of their functionality. For example, NetRanger used HP OpenView as its user interface. NetStalker could send messages to other network management stations via an SNMP trap message. Both the *JiNao Report*[142] and *Emerald 1997*[143] explicitly mentioned using SNMP traffic as a data source for their intrusion detection monitors. In addition, all of the MIB and RMON metrics are designed

[140] RFC 1157, A Simple Network Management Protocol (SNMP), May 1990 [SYM_P_0501113- SYM_P_0501142] and [SYM_P_0527111]; RFC 1155, Structure and Identification of Management Information for TCP/IP-based Internets, May 1990 [SYM_P_0501012- SYM_P_0501031]; RFC 1213, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991 [SYM_P_0501143- SYM_P_0501205]; RFC 1271, Remote Network Monitoring Management Information Base, November 1991 [SYM_P_0501206- SYM_P_0501271]; RFC 2021, Remote Network Monitoring Management Information Base Version 2 using SMIv2 , January 1997 [SYM_P_0603708- SYM_P_0603837].

[141] W. Stallings, SNMP, SNMPV2, SNMPV3 AND RMON 1 AND 2, 3rd ed. 1999.

[142] *JiNao Report* at 5-6.

[143] *Emerald 1997* at 356.

for monitoring network entities such as routers and gateways. Many network attacks such as sweeps and SYN floods are detectable through changes in the standard MIB variables. In fact, almost every network traffic category represented in the patents was represented in the MIBs. Thus, network management applications such as HP OpenView and the RFCs that network management relied upon are directly relevant to the alleged inventions.

302. The network management effort was directed by the Internet Engineering Task Force (IETF), and was worked on by many different organizations and people over the years. The work began in the 1980s, and the first network management documents were published in 1988 (RFC 1067 – SNMPv1 protocol, RFC 1065 – SMIv1 language description, and RFC 1066 – MIB I). Hewlett-Packard developed the HP OpenView network management platform. The HP OpenView for Windows User Guide for Transcend Management Software Version 6.1, which was used for this report, was released in October of 1997 ("*HP OpenView manual*"). However, many other publications document the features of HP OpenView in the relevant timeframe.[144]

303. Network management in general and HP OpenView in particular was designed to help "manage large communications infrastructure."[145] Management included monitoring elements in the network (hosts, routers, bridges, network segments, etc.) for proper or expected behavior, alerting the appropriate entity (a person or another program) when improper or unexpected behavior was detected, and issuing commands to those network elements.

304. The network management paradigm is centered on the concept of a "managed node", a "management application," and a standardized interface. The

[144] *See, e.g.*, M. Miller, MANAGING INTERNETWORKS WITH SNMP, 2nd Ed., 1997.

[145] M. Rose, The Simple Book, An Introduction to Internet Management, 2nd Ed., 1994.

management application monitors, and if necessary, sends controlling information to the managed node. Because the protocols (SNMP) and management information base (MIB) description language (SMI) are standardized, a wide range of management applications and managed nodes can be mixed and matched. Furthermore, a management application can in turn be a managed node for a higher-level management application.

305. Salient features of network management systems prior to November 1997 included:

- Developed by a standards body (Internet Engineering Task Force (IETF)).
- The basic network management architecture provided a two-tiered hierarchy of management application and managed nodes.
- A single manager could monitor many managed nodes, and a single managed node could be managed by many network managers.
- The simple two-tiered network management hierarchy could easily be extended to multiple levels of hierarchy, and this was endorsed with the Manager-to-Manager MIB released in 1993.
- The IETF community endorsed a standard network monitoring capability with the RMON and RMON-II specifications.
- The RMON monitors could be run on stand-alone sensors, routers, or on network management platforms.
- RMON MIBs specified a wide range of subjects to monitor, including network segments as a whole, hosts, host-to-host communications, and communication for a particular application (applications are identified by information in the packet).
- RMON MIBs specified a wide range of network metrics to measure for each subject, including number of packets, number of bytes, and many error conditions.
- RMON sensors could be configured to monitor specific metrics and trigger alerts that were sent to higher-level monitors when unusual conditions are detected.
- HP OpenView provided a published API for third-party applications to interact with the OpenView network management application suite.

- HP OpenView provided a number of additional responses including visual alerts, audible alerts, starting additional programs, and sending a message to a pager.

306. Figure 12, which was scanned in from the 1997 book "Understanding SNMP MIBs" by Perkins and McGinnis,[146] shows the basic approach of network management systems. The basic "management station" and "managed node" can be scaled up to support very large networks. At Level 1 is the basic managed node (1). It receives messages from and transmits messages to the Level 2 manager, which in this case is a dual-role entity (2). As described in the text, the "Dual-role entities are typically used to forward SNMP messages (an SNMP proxy), or to consolidate and synthesize information from many systems and make that information available to a higher-level manager" (3). From our security perspective, the "consolidate and synthesize" means to correlate and integrate. The dual-entity object (2) can then forward the "consolidated and synthesized" information to a 3rd tier management station (4), which can also receive information directly from other managed nodes (5).

307. Figure 13 was scanned in from the 1993 book "SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards" by William Stallings[147] and was originally part of a 1992 article. Of particular interest for this report is remote monitoring, of which the management is specified by the RMON and RMON-II MIBs. Figure 13 shows an example of an RMON deployment for remote monitoring. In the lower left is a network router supporting an RMON monitor (1). At the bottom middle (2) and the bottom right (3) are PCs running RMON monitors. Each of these monitors must report to a higher-level network management station like the one at the top of the

[146] D. Perkins and E. McGinnis, UNDERSTANDING SNMP MIBS, 1997. (Level 1, 2, and 3 labels and the circled numbers have been added for this discussion).

[147] W. Stallings, SNMP, SNMPv2 and CMIP, The Practical Guide to Network-Management Standards, 1993.

figure (5). The top-level network management station is also running its own RMON sensor (5). Finally, on the far left is yet another RMON monitor (4), but this one not only reports to (5) but it also supports its own local management console.

308. Figure 14 captures several relevant aspects of RMON sensors. In the upper left corner is a sample of several of the subjects an RMON sensor monitored (1). Furthermore, the RMON MIBs describe a wide range of statistics that may be monitored for the various subjects, including the number of packets, bytes, broadcast packets, error messages, and protocol distributions (2).

309. Figure 14 also briefly shows RMON's approach to anomaly detection and alerting (3). The graph's vertical axis represents the number of packets for some subject (e.g., a host) (4). The RMON MIBs (and indeed, the general purpose MIB-II) is highly configurable as to what metrics are monitored for what subjects. The graph's horizontal axis represents time (5). For this measurement and subject, a normal range of values is determined (6), and an upper threshold (7) and a lower threshold (8) bound that normal range. The RMON sensor then monitors the measurement over time, and as long as the measurements remain in the normal range nothing happens. However, when the measurement first exceeds one of the thresholds of normal values (10), the RMON monitor automatically triggers an alert (11) that is sent to a higher-level monitor.

310. For the RMON sensors, the data source is network packets. For example, it can passively monitor all the packets on a network segment. Most network management agents can monitor additional details about the objects they monitor; for example, the standard MIB specification includes details about the state of a particular connection (i.e., a management application can check if a connection is in a half-open connection). Shown in Exhibit X is a chart of the different types of information that could be monitored via SNMP or RMON sensors.

311. Management applications receive messages from managed nodes or from other management stations. In the case of HP OpenView, individual management applications can be started by and receive messages from OpenView.

312. Any individual variable on a managed device can be polled to determine its current value, and the result of that value can be passed to management applications (more then likely, the management application requested that the value be polled). If the managed node observes a specific value for certain variables, it automatically alerts a network management station.

313. Beginning with the Manager-to-Manager Management Information Base (RFC 1451), the normal range of values for a variable can be defined and if the variable exceeds the normal range an alert is automatically sent to hierarchically higher monitors. HP OpenView extends this by providing support for detecting anomalous behavior for categorical variables too.

314. The network management infrastructure provides a wide range of responses to anomalous or suspicious behavior. At the simplest level, simple events (e.g., a network segment failing) can trigger an alert (via an SNMP trap) to be sent to higher-level monitors. Network management objects can also respond to more sophisticated behaviors by setting alarms to be triggered when a value exceeds a normal range.

315. Furthermore, network management applications are expected to be full-blown applications that can provide complex analysis and response. For example, HP OpenView provided a graphical user interface (GUI) that could automatically alert a user, sound an audible alarm, automatically execute additional programs, or even activate a remote paging device. Finally, network management systems that serve in a dual-role capacity could alert a hierarchically higher network management system. For example,

Perkins says dual-role entities are used to "consolidate and synthesize information from many systems and make that information available to a higher-level manager."[148]

316. As noted in Exhibit M, the *HP OpenView manual* specifically referenced and relied upon the noted RFCs. Furthermore, network management applications such as HP OpenView were publicly used with all of the noted RFCs. In the alternative, given the explicit direction in the *HP OpenView manual* to use the RFCs for network management, such a combination would have been obvious.

317. In my opinion, as more fully reflected in Exhibit M to my Report, the system disclosed in *HP OpenView manual* satisfied every limitation of the indicated claims of the '203 and '615 patents, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the asserted claims of the '203 and '615 patents (except '615 claims 7 and 64-73) were not novel when filed.

318. Furthermore, network management applications such as HP OpenView were in public use with the RFCs for network management prior to November 9, 1997, and this public use also anticipates the indicated claims.

319. In addition, as reflect in Exhibit M, it would have been obvious to combine the statistical detection method from the *Feather Thesis* with the HP OpenView system and Internet Standards, for the reasons discussed previously regarding the *Feather Thesis*, and thus claim 7 of the '615 patent is invalid due to obviousness. Given the fact that HP OpenView managed network infrastructure, and a firewall is network infrastructure, it would have been obvious to treat a firewall as a managed node. Thus, '615 claim 64-73 are invalid due to obviousness.

[148] Understanding SNP MIPs, p. 7.